

Example:

Your own ActiveSupport::Concern

In this video we will build our own simplified version of "ActiveSupport::Concern" module and then go through the original implementation code together and understand its code.

Task:

Implement Rails' your own module that imitates " ActiveSupport::Concern" module.

Background: ActiveSupport::Concern

```
module M
  def self.included(base)
    base.extend ClassMethods

    base.class_eval do
      scope :disabled, -> {
        where(disabled: true)
      }
    end
  end

  module ClassMethods
    # ...
  end
end
```

```
require 'active_support/concern'

module M
  extend ActiveSupport::Concern

  included do
    scope :disabled, -> {
      where(disabled: true)
    }
  end

  class_methods do
    # ...
  end
end
```

Custom::Concern

```
module MyModule
  include Custom::Concern

  def instance_method
    "instance method"
  end

  included do
    attr_accessor :foo
  end

  class_methods do
    def class_method
      "class method"
    end
  end
end

class MyClass
  include MyModule
end
```

```
MyClass.new.instance_method      # "instance method"
MyClass.new.instance_methods     # [[:foo, :foo=]]
MyClass.new.class_method         # "class method"
```

Custom::Concern

```
module Custom
  module Concern
    end
end

module MyModule
  include Custom::Concern

  def instance_method
    "instance method"
  end
end

class MyClass
  include MyModule
end

MyClass.new.instance_method
```

```
# "instance method"
```